



LOOPiN

주제

"공동체와 화합 : 더 나은 대학생활을 위한 서비스"

취미

포트폴리오

모임

정보 접근성

주제

책

"공동체와 화합 : 더 나은 대학생활을 위한 서비스"

소속감

교류 플랫폼

소통

공동체 의식

취미

포트폴리오

정보 접근성

모임

주제

"공동체와 화합 : 더 나은 대학생활을 위한 서비스"

책

소속감

소통

교류 플랫폼

공동체 의식

대학생들을 위한 실명 기반 취미 모임 플랫폼

대학생들을 위한
실명 기반 취미 모임 플랫폼

학교 인증 기반 실명 활동

신뢰 기반 연결 유도

모임 중심 플랫폼

커뮤니티가 아닌 모임 단위로 활동

가벼운 참여 구조

동아리처럼 부담X, BUT 목적 중심적

→ '모임'단위 중심으로 설계되어
익명 커뮤니티와 달리
책임 있는 소통+일회성을 넘는 관계 형성

핵심기능

01

같은 학교 학생끼리 모임을 나눌 수 있어요.

02

취미나 목표에 맞는 모임을 만들고 참여할 수 있어요.

03

AI가 나에게 맞는 모임을 추천해줘요.

04

학생증 인증으로 학교 소속을 확인해요.

Point Color

#FF6BC4

→ 공동체의 활기, 에너지, 따뜻한 감성을 강조



루핀(LOOPIN)

loop in :참여시키다 / 연결하다



공동체 속 연결과 소통을 핵심 가치로 삼음

로고 디자인은 이러한 철학을 시각적으로 표현하는데 중점

루핀(LOOPIN)

loop in :참여시키다 / 연결하다

중앙의 'OO'를 사슬처럼 연결된 원형으로 구성

→ 사람과 사람 사이의 연결, 지속적인 네트워크, 그리고 연대의 상징



전체적으로 동글동글한 곡선형 디자인

→ 부드럽고 친근한 인상을 주며, 공동체의 따뜻하고 유연한 분위기를 표현

루핀 (LOOPIN)

loop in :참여시키다 / 연결하다

→ '연결'과 '순환'의 이미지를 자연스럽게 로고에 녹여내고자 함

시연

AI 기반 추천

OCR 학생증 인식

클라우드 활용

JWT 인증

AI 기반 추천

자유로운 대화로 관심사 입력 시, 적합한 모임 추천
모델 기반 자연어 처리로 사용자의 의도를 유연하게 파악 가능

```

44 def create_template_message(json):
52     day_time_string = ""
53     if day_and_time is not None:
54         day_and_time = day_and_time.split(",")
55         for elem in day_and_time:
56             elem.strip()
57             day_list = day_and_time[0:len(day_and_time) - 1]
58             time = day_and_time[len(day_and_time) - 1].split(" ")
59             for elem in day_list:
60                 day_time_string += DAYS[elem] + ", "
61             day_time_string += time[0][:2] + ":" + time[0][2:] + " - " + time[1][:2] + ":" + time[1][2:]
62     else:
63         day_time_string = "미정"
64
65     input_msg = (f"1. ID = {club['id']}\n"
66                 f"2. Name: {club['name']}\n"
67                 f"3. Category: {club['category']}\n"
68                 f"4. Subcategory: {club['subCategory']}\n"
69                 f"5. Description: {club['description']}\n"
70                 f"6. Date/Time: {day_time_string}\n"
71                 f"7. Location: {club['whereMeet']}\n"
72                 f"8. Maxmember: {club['maximumMember']}")
73     msg_list.append(HumanMessage(content = input_msg))
74
75     msg_list.append(HumanMessage("이상으로 모든 모임에 대한 정보를 설명했습니다. 이제 제가 원하는 모임에 대
76     msg_list.append(MessagesPlaceholder("UserInput"))
77     msg_list.append(HumanMessage(TASK_DESCRIPTION))
78     return msg_list
    
```


OCR 학생증 인식

학생증 이미지 업로드 시, 학생 정보 자동 추출
OCR 기술을 활용 → 학교·학과·학번 정보 자동 입력

```

84 class OCR:
85     def __init__(self):
86         self.reader = easyocr.Reader(['en', 'ko'], gpu=torch.cuda.is_available())
87
88     def run(self, src: np.ndarray) -> tuple[bool, dict]:
89         success = False
90         result = self.reader.readtext(src, detail = 1)
91         university = University.UNKNOWN
92         for i, elem in enumerate(result):
93             text = elem[1]
94             intersection = len(set.intersection(*[set("KYUNG HEE"), set(text)]))
95             union = len(set.union(*[set("KYUNG HEE"), set(text)]))
96             jaccard_similarity = intersection / float(union)
97             if jaccard_similarity > 0.75:
98                 university = University.KYUNGHEE
99                 break
100
101         # 이미지 전처리: 회색조 변환, 가우시안 블러, 캐니 엣지 검출
102         gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
103         blurred = cv2.GaussianBlur(gray, (3, 3), 0)
104         edged = cv2.Canny(gray, 75, 200)
105
106         # �츠 알고리즘 이진화 후 윤곽선 검출
107         ret, otsu = cv2.threshold(gray, -1, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)
108         contours, hierarchy = cv2.findContours(otsu, cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)
109
110         origin_area = src.shape[0] * src.shape[1]
111         min_area = origin_area * 0.5 # 최소 면적 설정
112         max_area = origin_area * 0.97 # 최대 면적 설정
113
114         # 설정한 면적 범위 안에서 외곽선 찾으면 크롭, 아니면 원본 이미지 전체를 학생증으로 인식
115         crop_img = src.copy()
116         for cnt in contours:
117             area = cv2.contourArea(cnt)
118             if min_area < area < max_area: # 학생증 영역을 찾기 위한 면적 범위
119                 x, y, width, height = cv2.boundingRect(cnt)
120                 crop_img = crop_img[y:y+height, x:x+width]
121                 break
122
123         # 신형 학생증이면 선명한 붉은톤, 구형 학생증이면 흰 톤
124         b, g, r = cv2.split(crop_img)

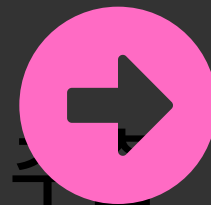
```


×

학생증 인증

본인명의로의 학생증을 업로드해주세요

학생증 업로드



×

학생증 인증

본인명의로의 학생증을 업로드해주세요



×

학생증 인증

본인명의로의 학생증을 업로드해주세요



학생 정보 확인

* 학교

경희대학교

* 학과

소프트웨어융합학과

* 학번

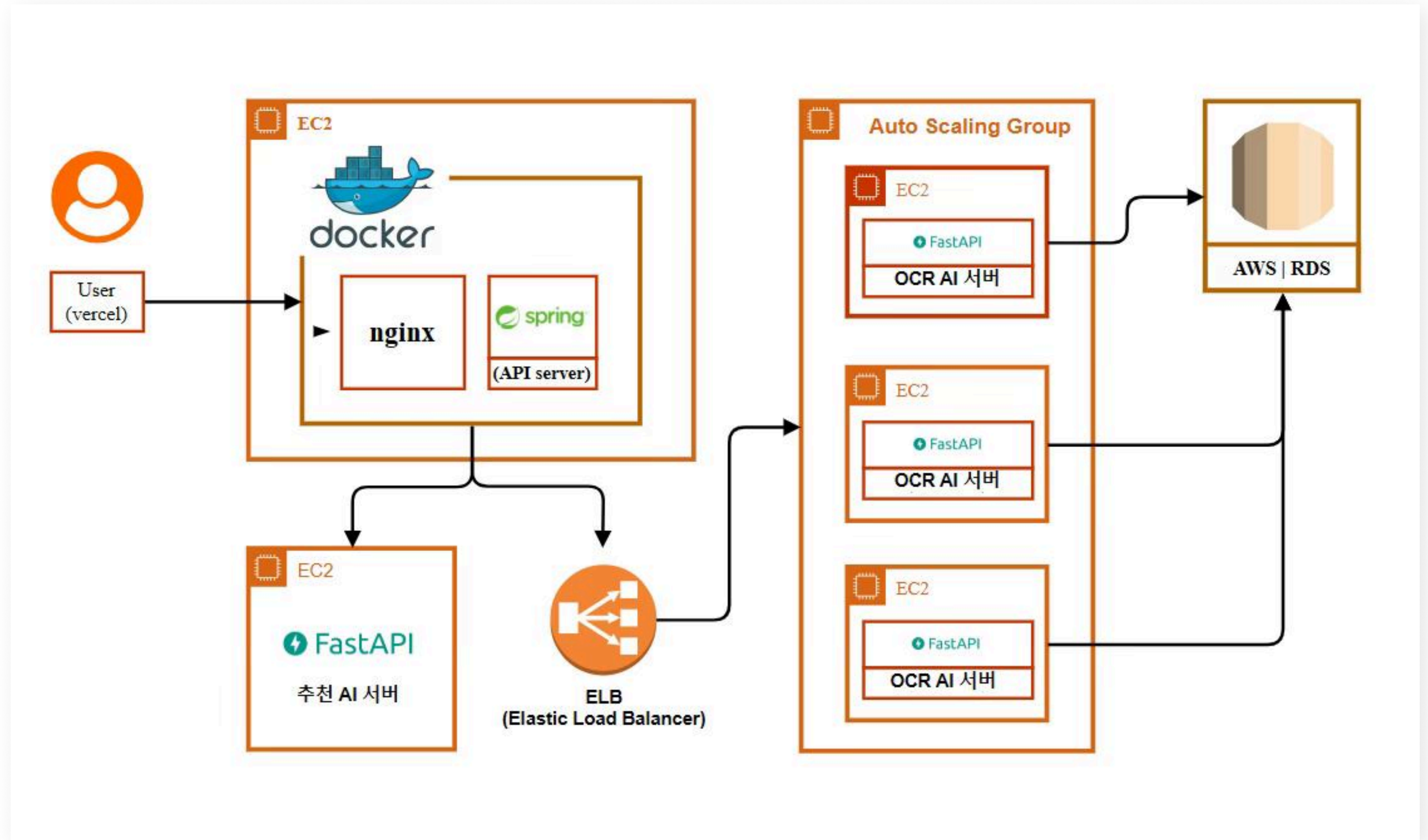
2019102137

다음

클라우드 활용

Docker + Nginx 기반 API 서버와
GPU 기반 AI 서버 클라우드 배포

OCR 서버는 자동 확장됨.
로드밸런서로 트래픽 안정적 분산



JWT 인증

JWT 기반 인증으로 로그인 후에도 사용자 정보를 안전하게 유지
세션 없이도 인증 가능한 구조. 확장성과 보안성을 확보

2025-04-06

LOPin

감사합니다.

Q&A